Location based Solar Tracking Device

Group # SD1208
Alan Grossman
Nick Blonigen
Client: John Ihle
Advisor: Mark Schroeder

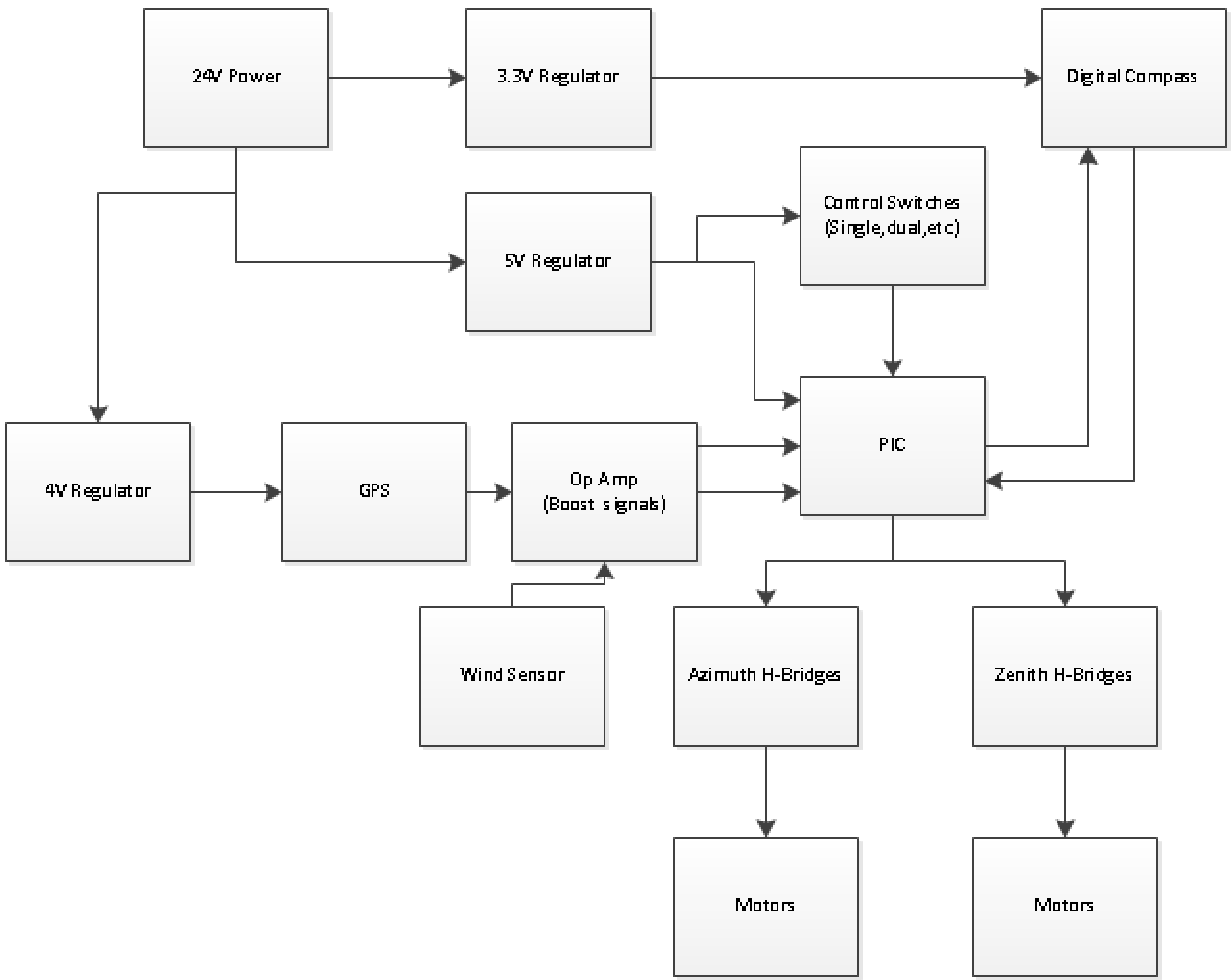
Introduction

John Ihle is our locally based client who wants a location based solar tracker. This involves designing and developing a controller that can be retrofitted to a dual-axis solar photovoltaic tracker system.

Client Requirements

- Currently the system will have to be compatible with both single and dual axis panels
- Current Power that has to be compatible with is a 24V system
- The Panels Have the capability to go 85 degrees to flat and no limit on the horizontal axis rotation
- Automatic and manual modes desired
- The ability to switch between single and dual axis mode also desired
- High grade equipment is desired to withstand extreme temperature conditions

Block Diagram



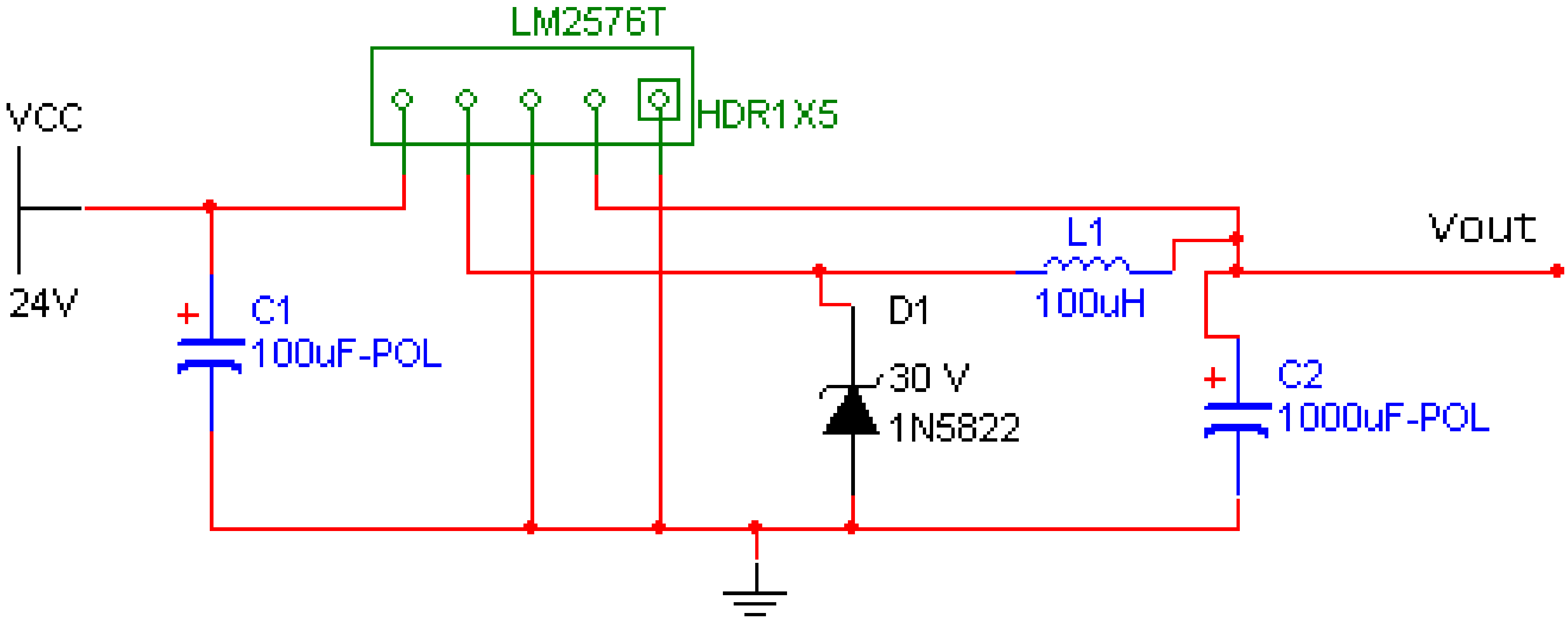
5V Regulator

- LM2576/LM2576HV Series - SIMPLE SWITCHER® 3A Step-Down Voltage Regulator

- 24V to 5V LM2576T-5.0-ND
- Advantages:
- The regulator guarantees a 3A output current
- Wide input range to compensate for power fluctuations
- High efficiency
- Thermal shutdown protection
- Fixed frequency internal oscillator
- High Temp Thresholds
- Disadvantages:
- Output Voltage Ranges from 4.8-5.2V

We used this part to bring our 24V power supply down to 5V useable power supply for our components.

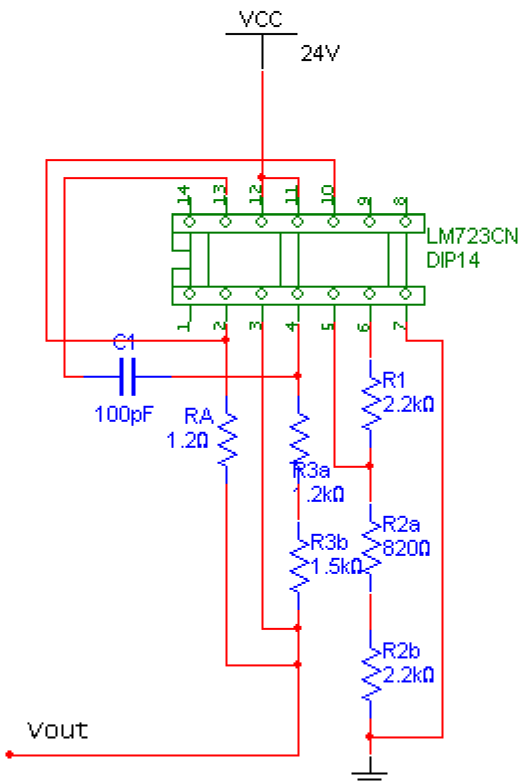
5V Schematic

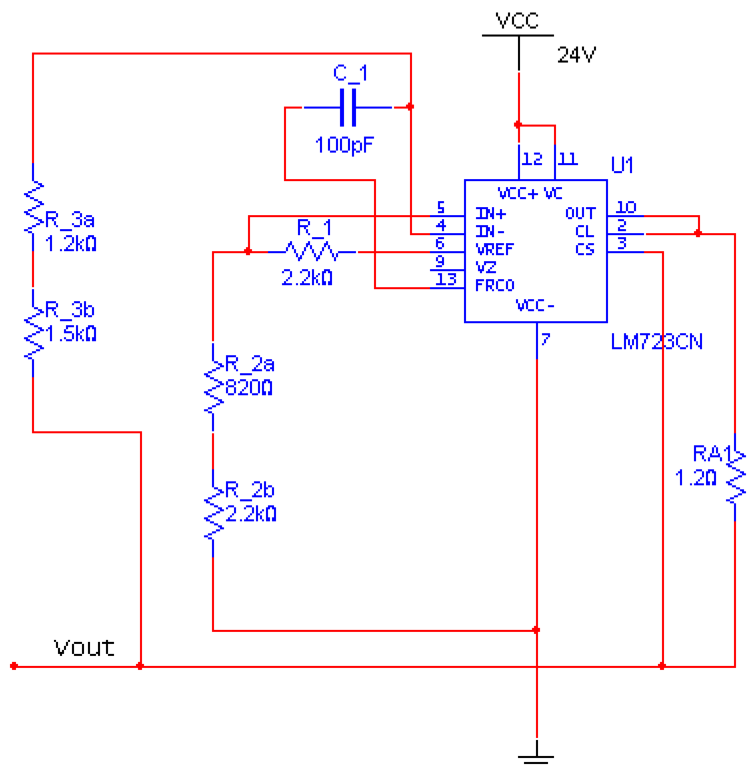


4V and 3V Regulators

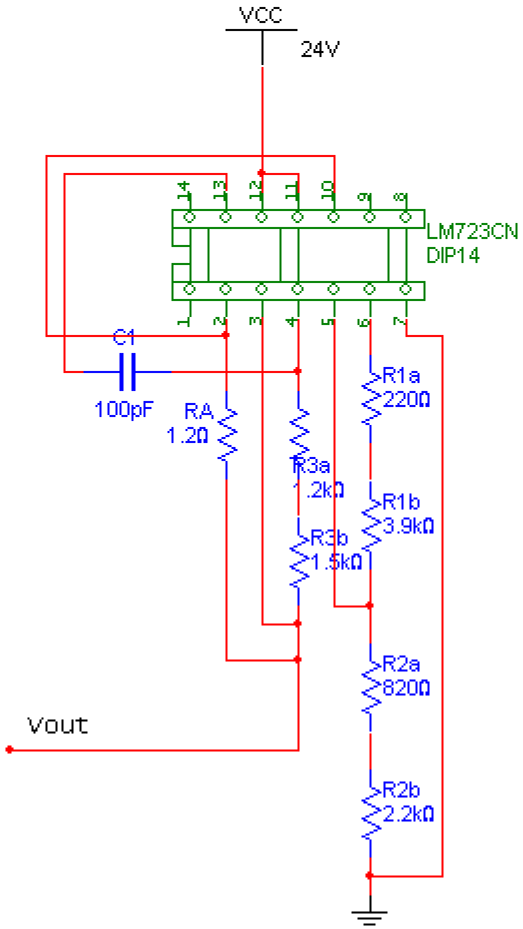
- LM723
- -40 to 125C
- 150mA output
- Adjustable voltage
- Precision voltage
  - GPS needs this to be accurate
- Used with digital compass because we were already using one for GPS

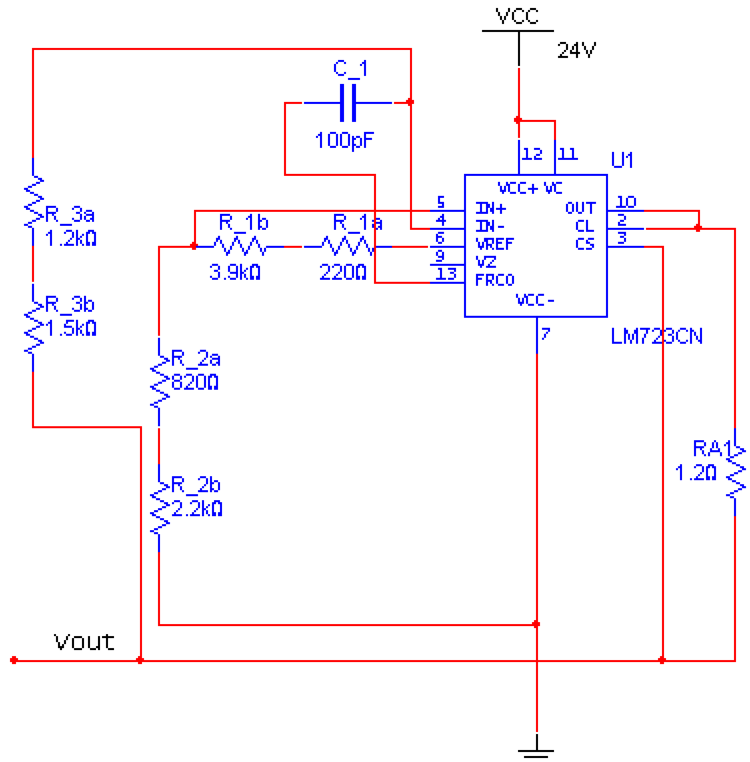
4V Schematic





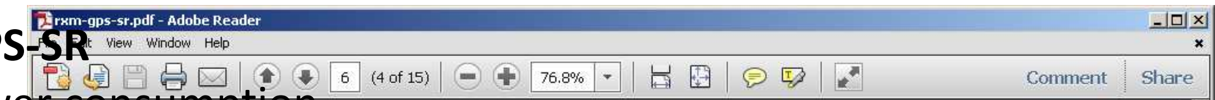
3V Schematic





GPS Sensor

- **RXM-GPS-SR**
- Low power consumption (46mW)
- High sensitivity (159dBm)
- No programming necessary
- Includes Date and Time



POWER CONTROL

By default, the SR Series will operate in full power mode. However, it also has a built-in power control mode called Adaptive Trickle Power mode. In this mode, the receiver will power on at full power to acquire and track satellites and obtain satellite [illegible] the RF stage and only uses its processor stage (CPU) to determine a position fix (which takes about 160mS). Once the fix is obtained, the receiver goes into a low power standby state. After a user-defined period of time, the receiver wakes up to track the satellites for a user-defined period [illegible] using the CPU only, and then resumes stand-by. The initial acquisition time is variable, depending on whether it is a cold start or assisted, but a maximum acquisition time is definable. This cycling of power is ideal for battery-powered applications since it significantly reduces the amount of power [illegible] the receiver while still providing similar performance to the full power mode.

In normal conditions, this mode provides a fixed power savings, but under poor signal conditions, the receiver returns to full power to improve performance. The receiver sorts the satellites according to signal strength and if the fourth satellite is below 26dB-Hz, then the receiver switches to full power. Once the fourth satellite is above 30dB-Hz, the receiver returns to Adaptive Trickle Power mode.

For optimum performance, SiRF recommends cycle times of 300mS track to 1S interval or 400mS track to 2S interval. CPU time is about 160mS to compute the navigation solution and empty the UART. There are some situations in which the receiver stays in full power mode. These are: to collect periodic ephemeris data, to collect periodic ionospheric data, to perform RTC convergence, and to improve the navigation result. Depending on states of the power management, the receiver will be in one of three system states:

Full Power State

All RF and baseband circuitry are fully powered. There is a difference in power consumption during acquisition mode and tracking mode. Acquisition requires more processing, so it consumes more power. This is the initial state of the receiver and it stays in this state until a reliable position solution is achieved.

CPU Only State

This state is entered when the satellite measurements have been collected but the navigation solution still needs to be computed. The RF and DSP processing are no longer needed and can be turned off.

Stand-By State

In this state, the RF section is completely powered off and the clock to the baseband is stopped. About 1mA of current is drawn in this state for the internal core regulator, RTC and battery-backed RAM. The receiver enters this state when a position fix has been computed and reported.

Page 6

TYPICAL APPLICATIONS

Figure 4 shows a typical application for the module.

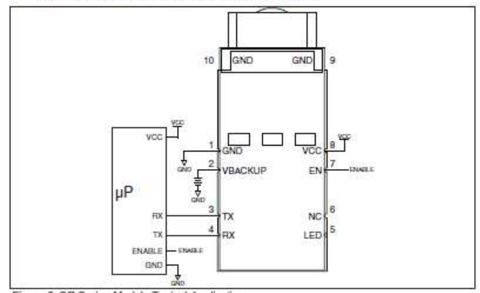


Figure 3: SR Series Module Typical Application

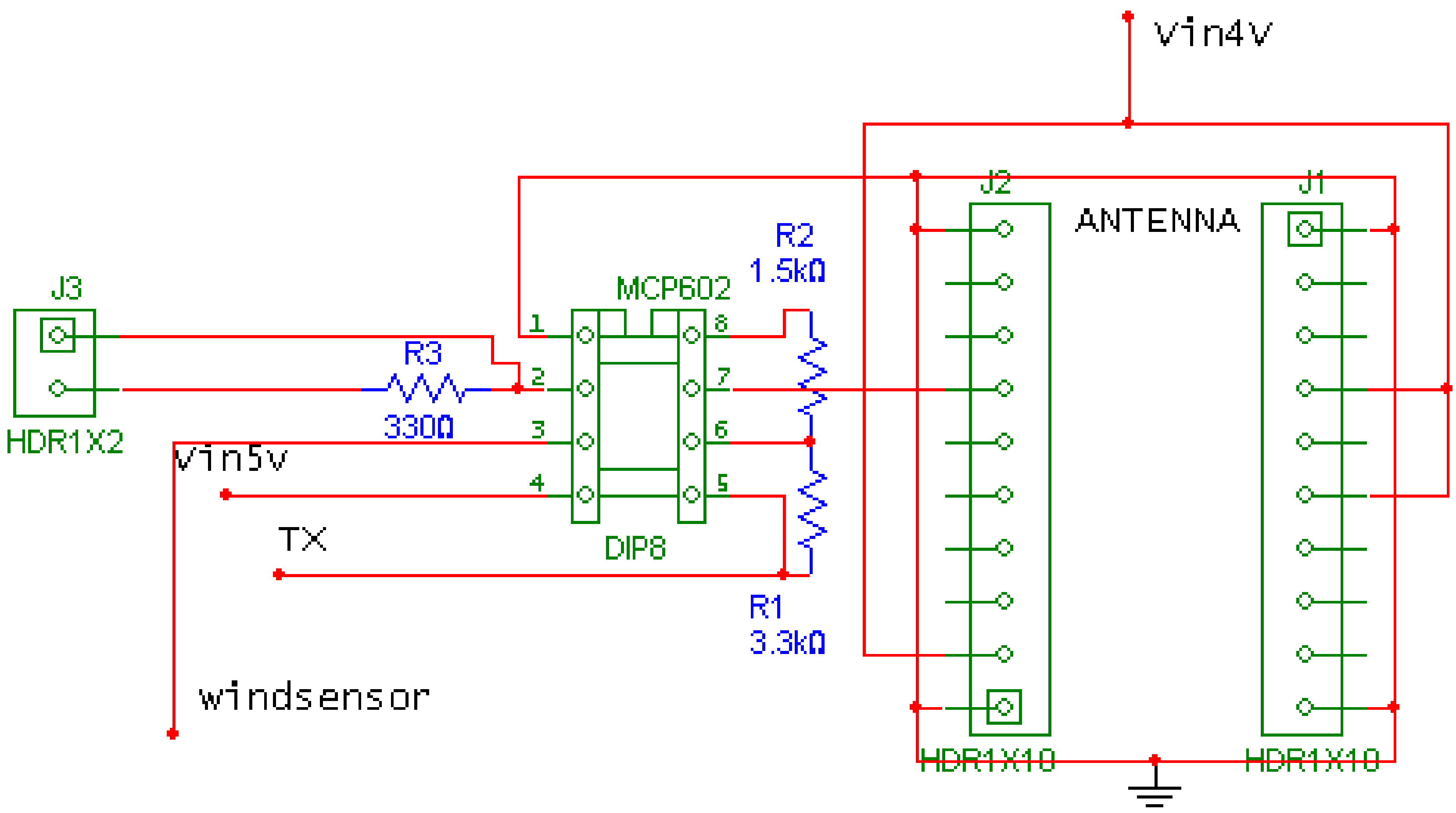
SLOW START TIME

The most critical factors in start time are current ephemeris data, signal strength, and sky view. The ephemeris data describes the path of each satellite as they orbit the earth. This is used to calculate the position of a satellite at a particular time. This data is only usable for a short period of time, so if it has been more than a few hours since the last fix or if the location has significantly changed (a few hundred miles), then the receiver may need to wait for a new ephemeris transmission before a position can be calculated. The GPS satellites transmit the ephemeris data every 30 seconds. Transmissions with a low signal strength may not be received correctly or be corrupted by ambient noise. The view of the sky is important because the more satellites the receiver can see, the faster the fix and the more accurate the position will be when the fix is obtained.

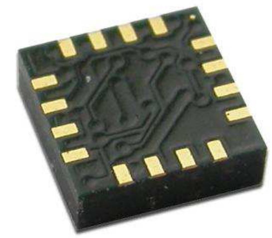
If the receiver is in a very poor location, such as inside a building, urban canyon, or dense foliage, then the time to first fix can be slowed. In very poor locations with poor signal strength and a limited view of the sky with outdated ephemeris data, this could be on the order of several minutes. In the worst cases, the receiver may need to receive almanac data, which describes the health and course data for every satellite in the constellation. This data is transmitted every 15 minutes. If a lock is taking a long time, try to find a location with a better view of the sky and fewer obstructions. Once locked, it is easier for the receiver to maintain the position fix.

Page 7

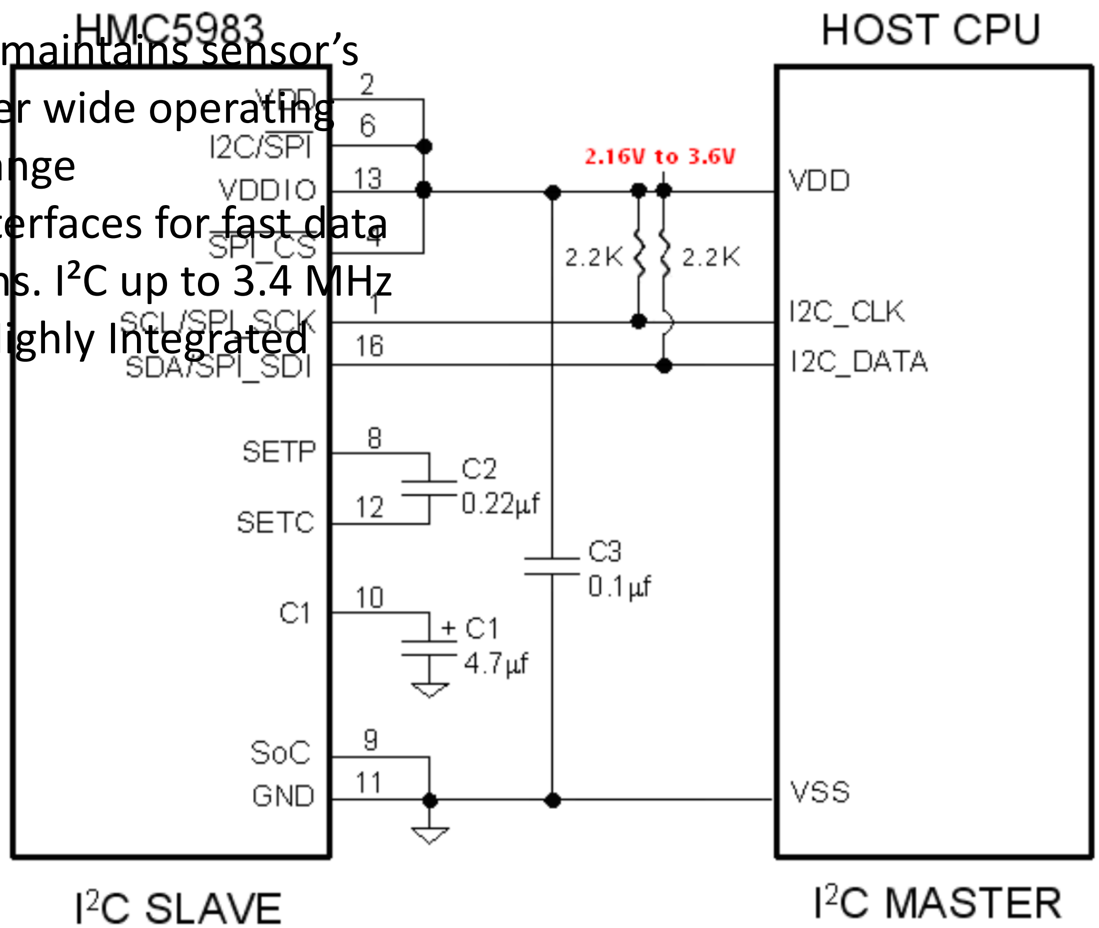
GPS Schematic



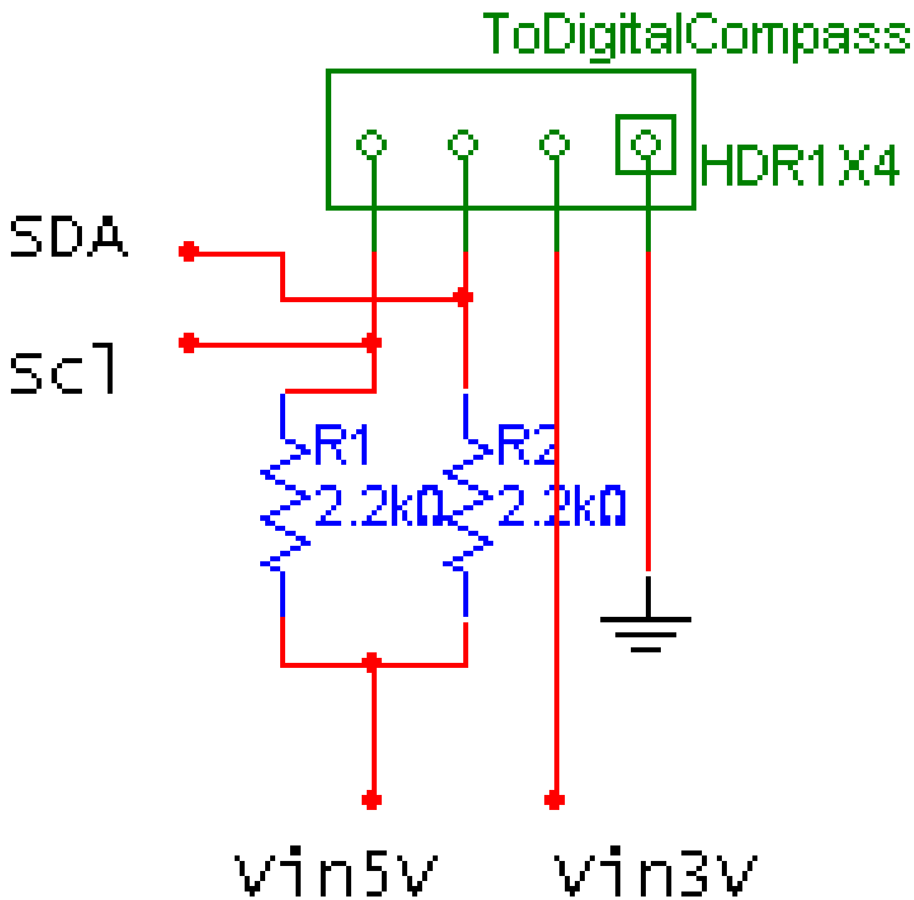
Digital Compass



- Automatically maintains sensor’s sensitivity under wide operating temperature range
- High-speed interfaces for fast data communications. I^2C up to 3.4 MHz
- Small size for Highly Integrated Products



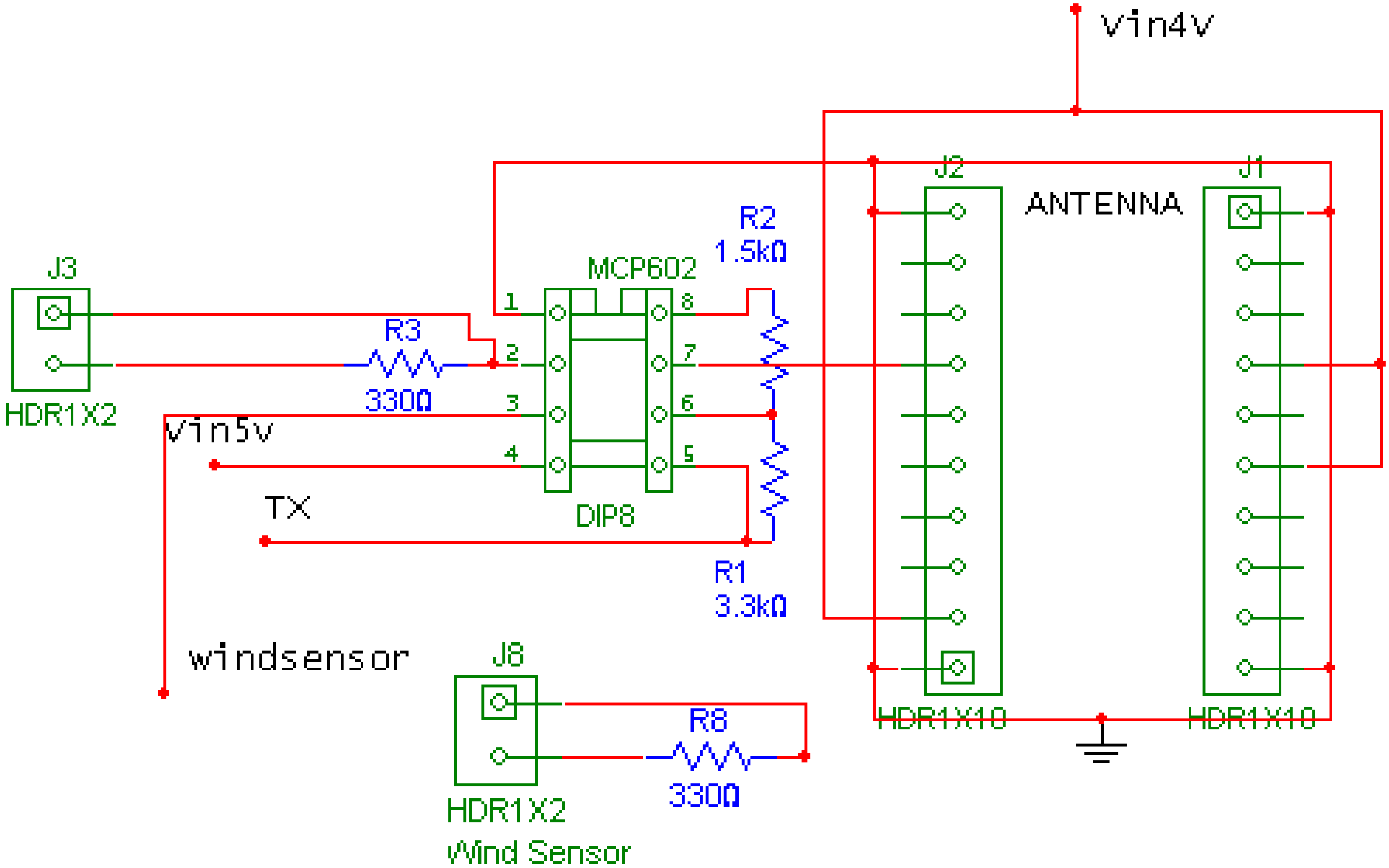
Digital Compass Schematic



Wind Sensor

- John wanted this added in the second semester
- He had a bunch of these laying around and gave us it
- 3-cup anemometer
- 1 m/s to 96 m/s (2.2 mph to 215 mph) (highest recorded)
- Consensus Transfer Function
- 0 Hz to 125 Hz (highest recorded)
- -55 °C to 60 °C

Wind Sensor Schematic



PIC Microprocessor

- Microchip PIC18F4620

Advantages:

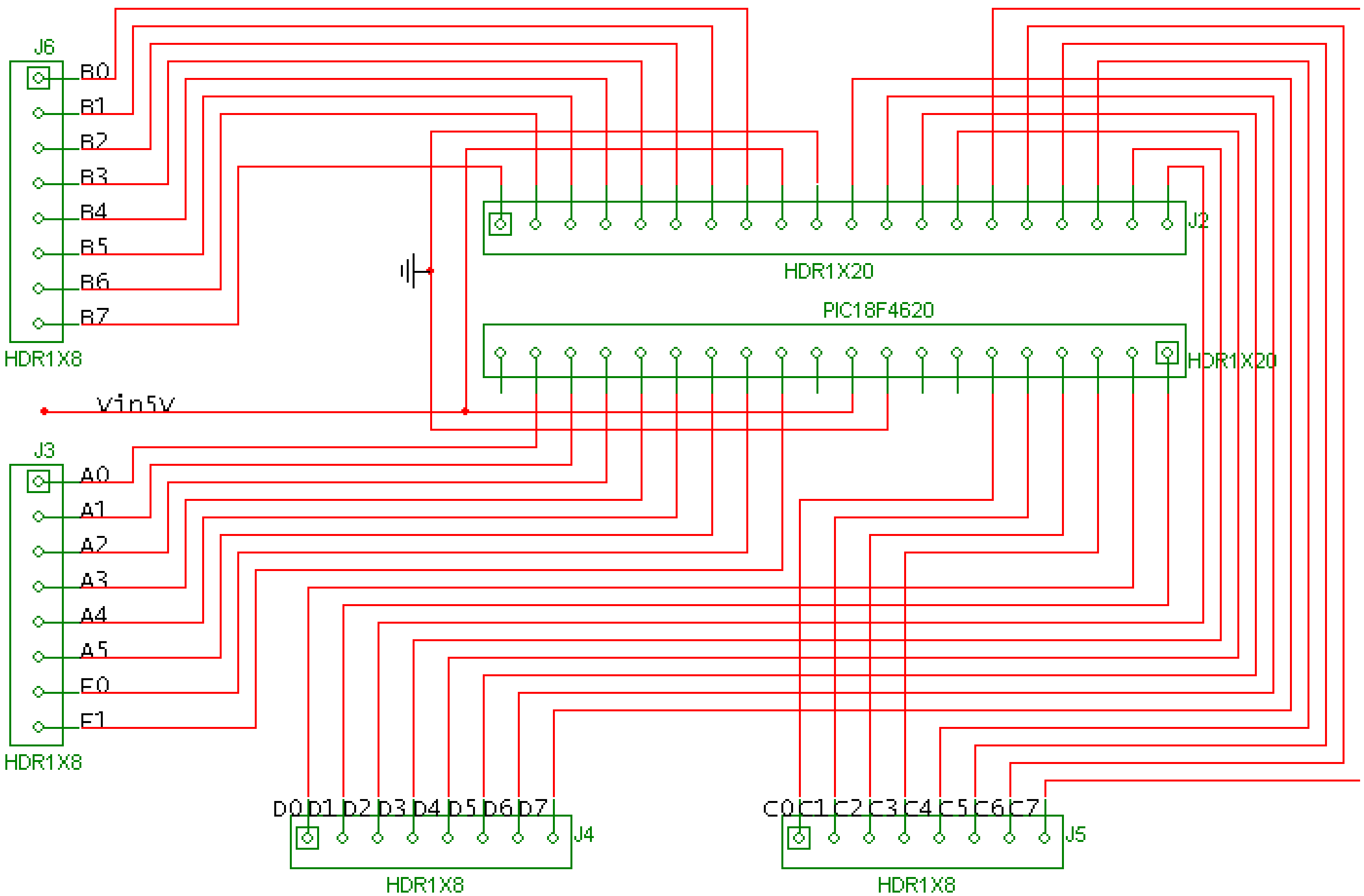
- Used before in class Familiar with microcontroller
- Comes with a compiler
- Has enough ports to handle our devices
- Has a boot loader
- Cost effective
- Optimized C language Compiling
- -40C to 125C

Disadvantages

- In comparison to other chips it is fairly large in size with low memory space

PIC Schematic

- Only shows picture of port connections



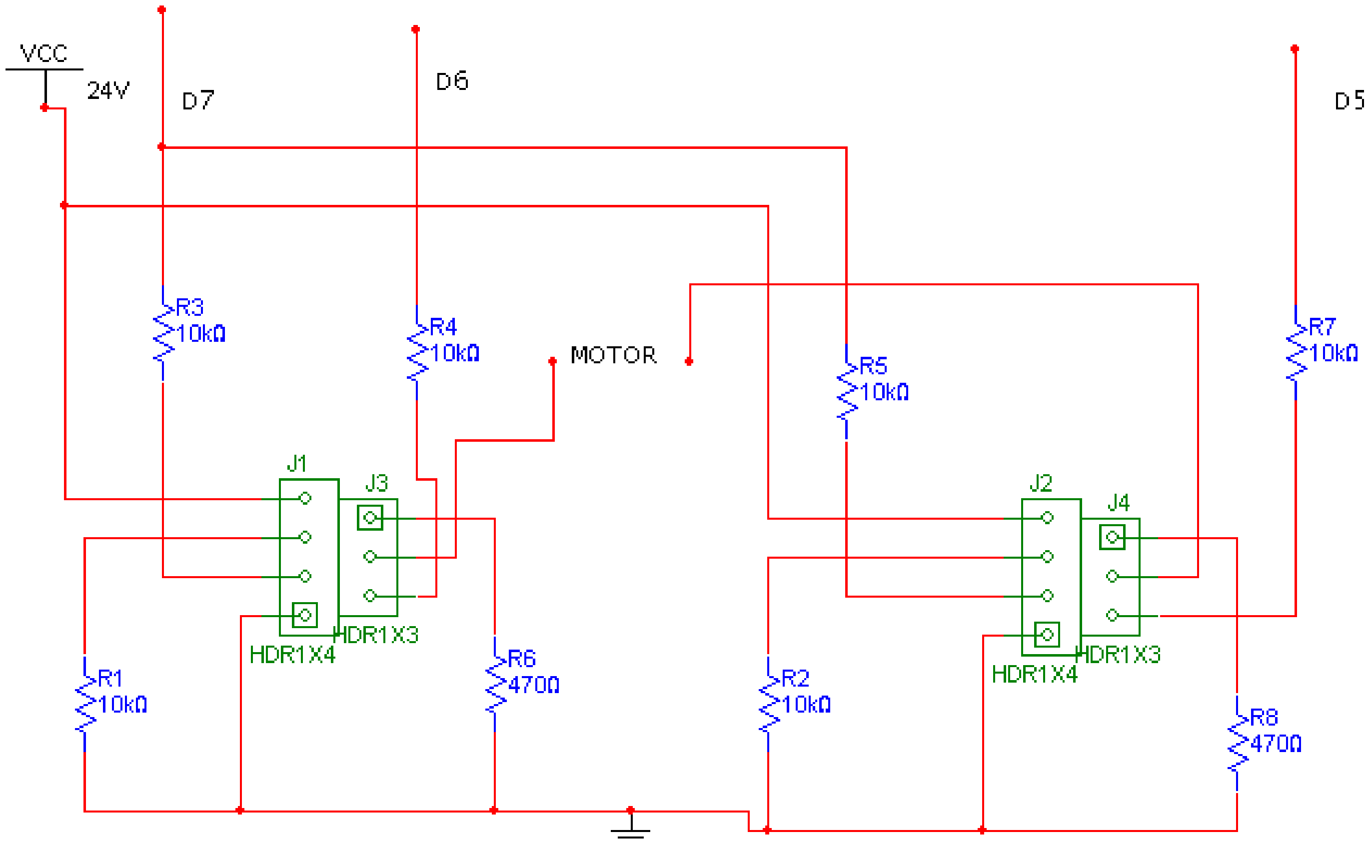
Motors

- Motors used in his panels
- 14954659 - VW76 24-36 VDC Actuator
- 24-36V DC Actuator
- 1/15HP
- Gear Ratio 19:1
- Current 3 Amps @ Full Load

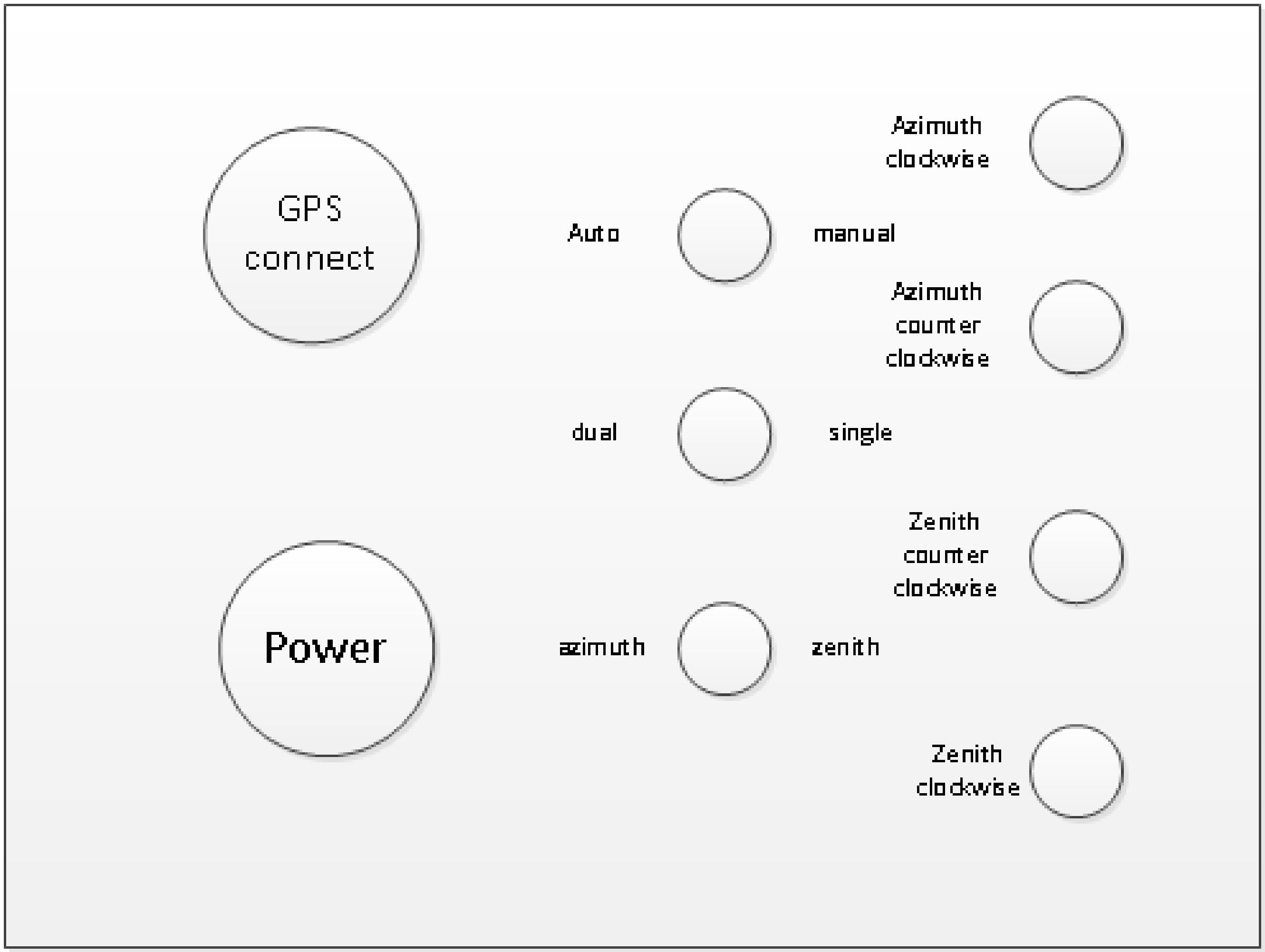
Half H-bridges

- John wanted to be able to use motors with 10A when we met with him second semester
- Had to change from 2 full 3A H-bridges to 4 10A half H bridges to meet requirement
- Used BTN7960,High Current PN Half Bridge
- -40 °C to +150 °C

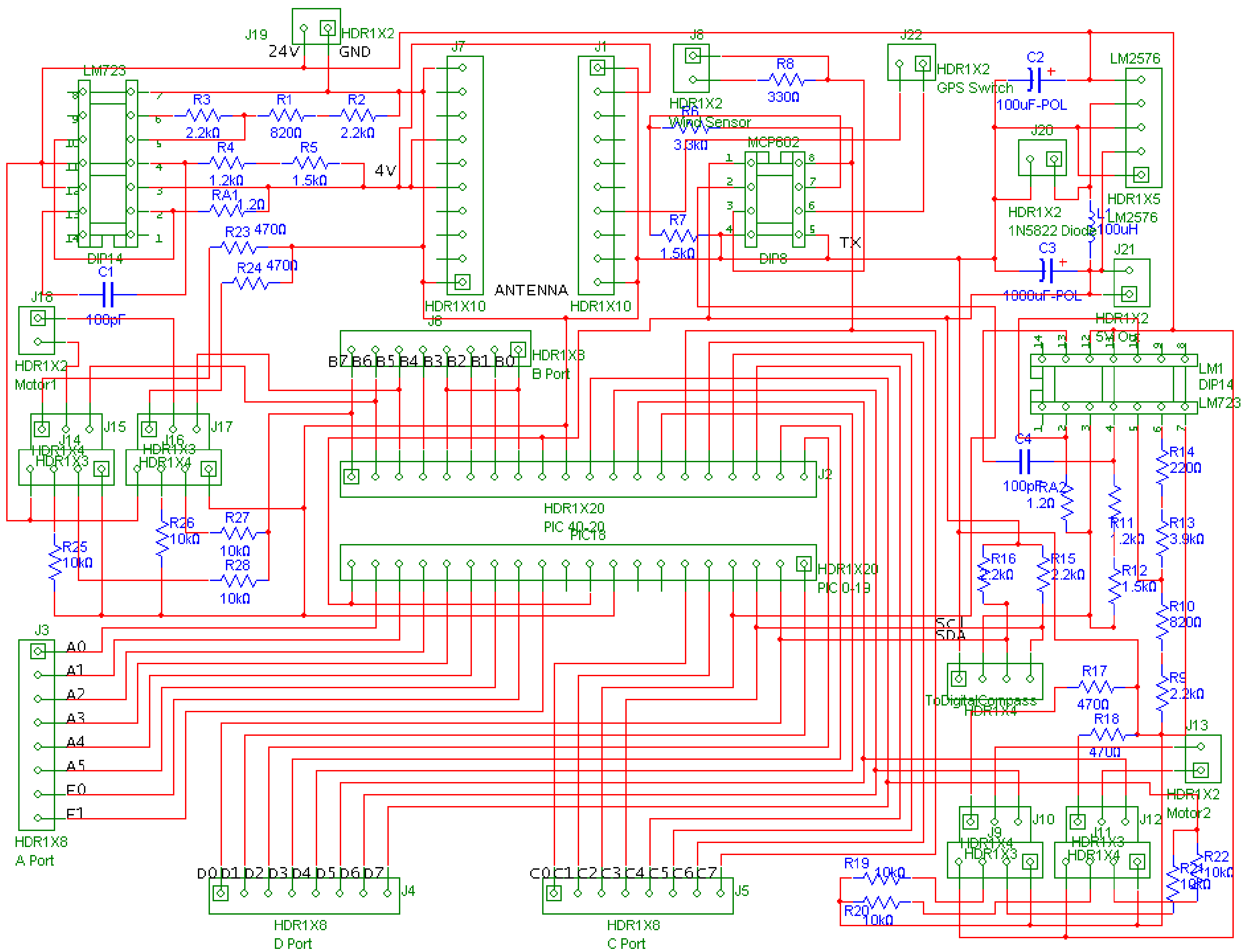
H-Bridge Schematic



Control Switches

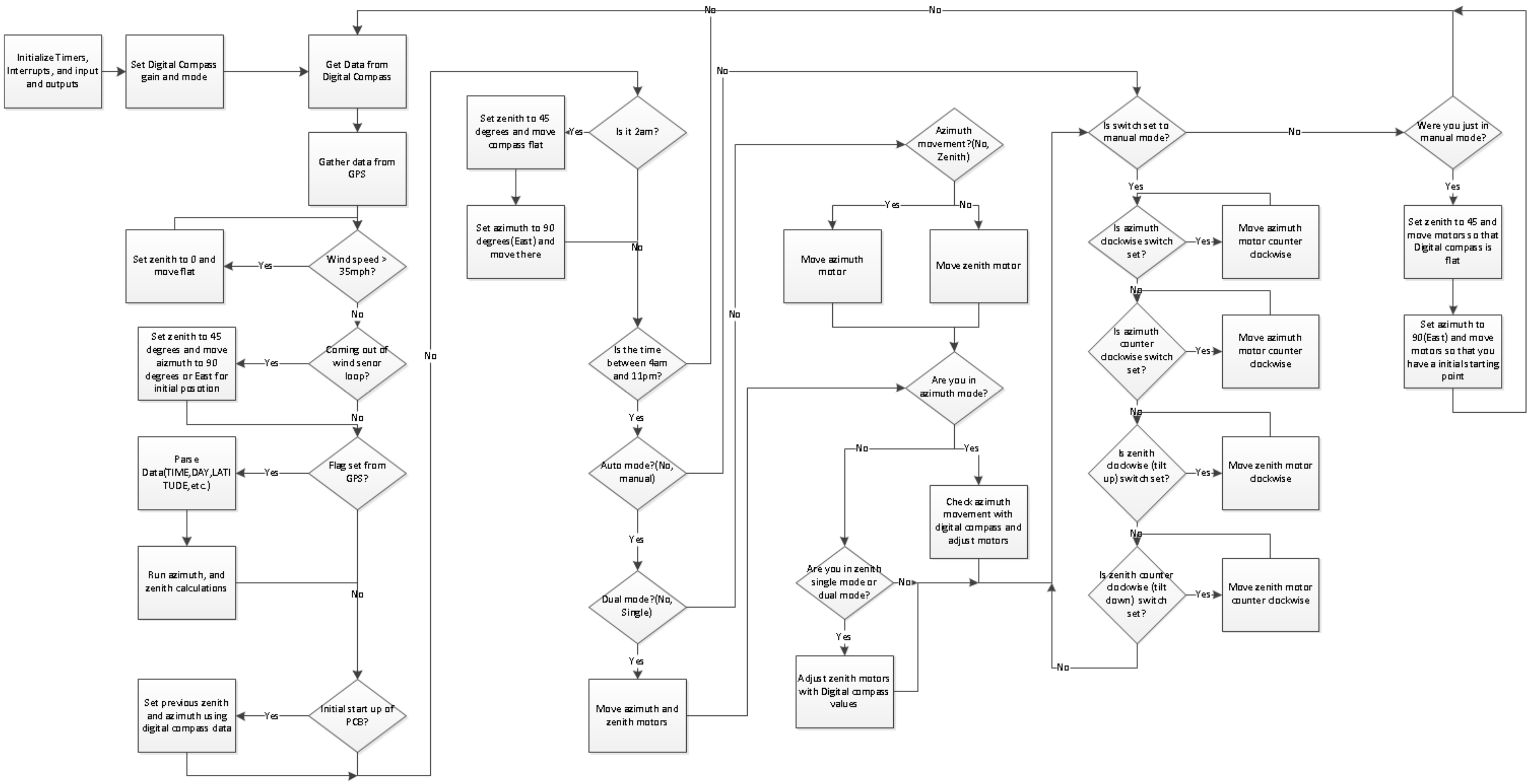


Schematic



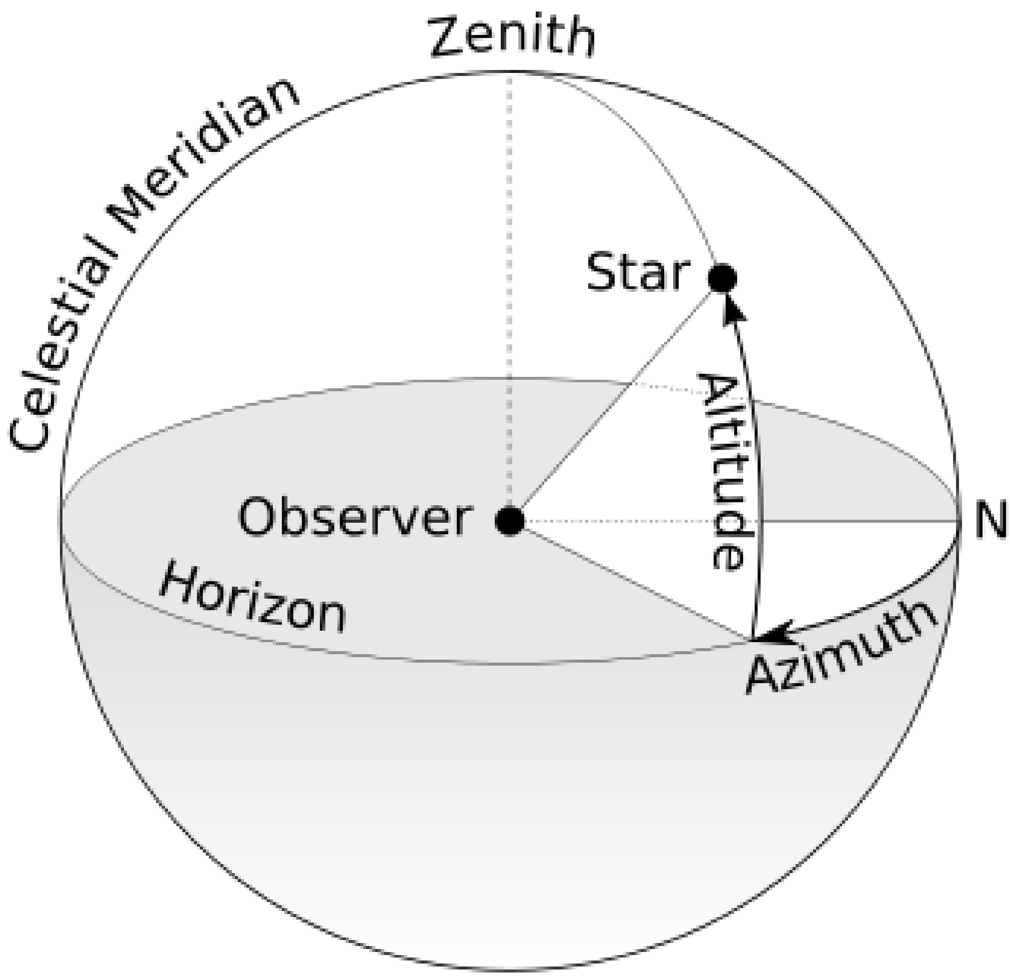
J19
HDR1X2
24V
GND
LM723
DIP14
R3
2.2kΩ
R1
820Ω
R2
2.2kΩ
R4
1.2kΩ
R5
1.5kΩ
4V
RA1
1.2Ω
R23 470Ω
R24 470Ω
C1
100pF
J18
HDR1X2
Motor1
J7
HDR1X10
J1
HDR1X10
ANTENNA
J8
HDR1X2
Wind Sensor
R8
330Ω
R6
3.3kΩ
MCP602
DIP8
R7
1.5kΩ
TX
J22
HDR1X2
GPS Switch
C2
100uF-POL
LM2576
HDR1X5
J20
HDR1X2
1N5822 Diode
L1
100uH
C3
1000uF-POL
J21
HDR1X2
5V Out
J6
HDR1X8
B Port
B7 B6 B5 B4 B3 B2 B1 B0
LM1
DIP14
LM723
J14
J15
J16
J17
HDR1X3
HDR1X4
J2
HDR1X20
PIC 40-20
PIC18
HDR1X20
PIC 0-19
R25
10kΩ
R26
10kΩ
R27
10kΩ
R28
10kΩ
C4
100pF
RA2
1.2Ω
R14
220Ω
R11
1.2kΩ
R13
3.9kΩ
R16
2.2kΩ
R15
2.2kΩ
R12
1.5kΩ
R10
820Ω
R9
2.2kΩ
SCL
SDA
ToDigitalCompass
HDR1X4
R17
470Ω
R18
470Ω
J13
HDR1X2
Motor2
J3
A0
A1
A2
A3
A4
A5
E0
E1
HDR1X8
A Port
J9
J10
J11
J12
R19
10kΩ
R20
10kΩ
R21
10kΩ
R22
10kΩ
D0 D1 D2 D3 D4 D5 D6 D7
J4
HDR1X8
D Port
C0 C1 C2 C3 C4 C5 C6 C7
J5
HDR1X8
C Port

Flow Chart



Initialize Timers, Interrupts, and input and outputs
Set Digital Compass gain and mode
Get Data from Digital Compass
Gather data from GPS
Wind speed > 35mph?
Set zenith to 0 and move flat
Coming out of wind senor loop?
Set zenith to 45 degrees and move aizmuth to 90 degrees or East for initial posotion
Flag set from GPS?
Parse Data(TIME,DAY,LATITUDE,etc.)
Run azimuth, and zenith calculations
Initial start up of PCB?
Set previous zenith and azimuth using digital compass data
Is it 2am?
Set zenith to 45 degrees and move compass flat
Set azimuth to 90 degrees(East) and move there
Is the time between 4am and 11pm?
Auto mode?(No, manual)
Dual mode?(No, Single)
Move azimuth and zenith motors
Azimuth movement?(No, Zenith)
Move azimuth motor
Move zenith motor
Are you in azimuth mode?
Check azimuth movement with digital compass and adjust motors
Are you in zenith single mode or dual mode?
Adjust zenith motors with Digital compass values
Is switch set to manual mode?
Is azimuth clockwise switch set?
Move azimuth motor counter clockwise
Is azimuth counter clockwise switch set?
Move azimuth motor clockwise
Is zenith clockwise (tilt up) switch set?
Move zenith motor clockwise
Is zenith counter clockwise (tilt down) switch set?
Move zenith motor counter clockwise
Were you just in manual mode?
Set zenith to 45 and move motors so that Digital compass is flat
Set azimuth to 90(East) and move motors so that you have a initial starting point
Yes
No

How to Determine Solar Position



To determine the solar position of any object in the sky we will need two angles solar azimuth and the solar zenith angles

Device Inputs needed

- Current Calendar Date (GPS)
- Time of Day (GPS)
- The device's longitude and latitude (GPS)
- Panel Face Direction (Digital Compass)

Calculation Variables for Solar Azimuth

- ϕ_s = Solar Azimuth angle
- Θ_s = Solar Elevation angle
- h = Hour Angle at present time
- δ = Sun declination
- Φ = Local Latitude

Solar Azimuth Angle

$$\phi_s = \cos^{-1}\left(\frac{\sin\delta - [\sin\Theta_s]\sin\Phi}{\cos\Theta_s\cos\Phi}\right)$$

$$\delta = \sin^{-1}\left[.39779\cos\left(.98565(N+10)\right) + 1.914\sin\left(.98565(N-2)\right)\right]$$

N= Number of days since midnight coordinated Universal Time as Jan 1 begins & can have decimals to adjust for local time zones.

$$\sin\Theta_s = \cos h\cos\Phi\cos\delta + \sin\delta\sin\Phi$$

ϕ_s = Solar Azimuth angle
Θ_s = Solar Elevation angle
h = Hour Angle at present time
δ = Sun declination
Φ = Local Latitude

Solar Zenith Angle

$$\cos \Theta_s = \sin \Phi \sin \delta + \cos \Phi \cos \delta \cos h$$

ϕ_s = Solar Azimuth angle
Θ_s = Solar Elevation angle
h = Hour Angle at present time
δ = Sun declination
Φ = Local Latitude

Testing and Evaluation

- We would first test it out on the breadboard, make sure it works and we would also know what values it should be giving
- Then we would move to PCB and test all the same items

Testing and Evaluation

- Set up circuits on breadboard
- Check for correct sensor data(GPS, digital compass, and wind sensor)
- Check Formula are working correctly
- Check Motor movement code
- Check all the mode manual, single, dual, and auto
- Check switches are working correctly to give voltage to PIC
- Test out on most of these items on the PCB

Budget:

Item #	Item Description	Price per Unit	Quanity	Total Cost
1	PIC Microcontroller	$30	2	$60
2	Precision Regulator	$3.00	4	$12
3	Voltage Regulators	$8	4	$32
4	GPS	$50	3	$150
5	Components	$60.00	1	$60
6	PCB	$66	1	$66
7	Digital Compass	$10.00	4	$40
8	Motors	$180	1	$180
9	H-bridges	$6.00	6	$36
			Total	$636

Timeline Semster1:

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Requirements Capture meeting					x											
Requirements Capture Document						x										
Options Considered							x									
Order Parts								x								
Make Schematic in Multisim									x							
Program PIC										x	x	x	x	x		
Breadboard PIC (maybe w/o sensors)										x	x	x				
Interface w/ sensors and motor																x
>>> Sensors											x	x	x			
>>> PIC											x	x	x			
>>> Motor																x
Progress Report:													x			
>>> Progress on PIC														x		
Prepare for presentation														x		
Presentations, end of semester															x	

Timeline Semster2:

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Requirements Capture Update	x													
Order Motor	x													
Order Parts	x	x												
Make Schematic in Multisim						x	x							
Program PIC		x	x	x	x	x								
Breadboard PIC (maybe w/o sensors)		x	x	x	x	x								
Interface w/ sensors and motor			x	x	x	x								
Progress Report:							x							
Revise Schematic in Multisim								x	x					
Start PCB layout in Ultiboard									x	x				
Finalize PCB layout (Ultiboard)										x	x			
Prepare for presentation											x	x	x	
Presentations, end of semester														x

Problems and Lessons Learned

- Time to get sensors to work took longer than expected
- Found out near the end that magnetometer need to be calibrated for soft and hard irons
  - Had to do some matlab code to get comp values
- If magnetometer was tilted, azimuth measurements would become wrong
  - Had to added code that would fix this
- PCB layout issues
- C code complier(lite mode)
  - Had to download a trial version to be able to build our code

Future work

- Use accelerometer so that digital compass could tilt. This would simply code a lot.
- Work calibration for digital compass into the code
- Tin can regulators on PCB

Summary

- We think it should all work, We tested it out what we could here.
- We still need to bring it out to John's place and do calibration there and motor movement under loaded condition might change part of code
- Then see if it will work for a period of time at his place
- We had to use matlab code to do calibration code to get rid of irons errors
- We had to added code to get around the fact that azimuth is messed up if digital compass is tilted